Dice Game Implementation

Part A: Problem Statement Specifications:

You are to implement a dice game in which the user rolls a pair of dice. The rules of the game are as follows:

1) The player rolls the dice and sums up the face values (upwards).
2) If the first roll sum is a 7 or 11, the player wins.
3) If the first roll sum is a 2, 3 or 12, the player looses.
4) If the first roll is any other number, that sum becomes the '**Player's Earned Point (PEP)'/'Point Match (PM)'**.
5) To win, the player must continue rolling the dice until him/her "**makes the highest PEP/PM**"

The following demonstrates a few executions of the program:

```
# ./02_Dice
                    ROLL THE DICE [ENTER]

---------------------------------------------------------------------
  ROLL NUM      DICE #1       DICE #2    TOTAL ROLL  POINT MATCH
---------------------------------------------------------------------
        1             2             6             8             8
        2             3             1             4             8
        3             6             2             8             8
Congratulations you roll 8 and WON

Another Game? [ Y / N ]  Y
                    ROLL THE DICE [ENTER]
---------------------------------------------------------------------
  ROLL NUM      DICE #1       DICE #2    TOTAL ROLL  POINT MATCH
---------------------------------------------------------------------
        1             4             3             7             7
Congratulations you roll 7 and WON at your first try!!!!
Another Game? [ Y / N ]  Y

                    ROLL THE DICE [ENTER]
---------------------------------------------------------------------
  ROLL NUM      DICE #1       DICE #2    TOTAL ROLL  POINT MATCH
---------------------------------------------------------------------
        1             1             1             2             2
Sorry, you roll 2 and you loose!!!
```

To terminate the program:

```
Another Game? [ Y / N ]N
Thank you for playing
You won 0 games and lost 3 games!
Better luck next time!

Another Game? [ Y / N ]N
Thank you for playing
You won 3 games and lost 0 games!
What a winner!
```

In order to successfully complete this program and obtain all the marks, you will need to:

1) Define WON and LOST as macros in your program. Use the values of 0 for WON and 1 for LOOSE
2) Implement a function, with function prototype int rollDice(void):
   - rollDice( ) should use rand( ) to randomly generate a number between 1 – 6 and should return the number generated by rand( )
3) Implement a function, with function prototype int playGame(void);
   - When the player is ready to play, (s)he would use the key ENTER to roll the dice
   - If the user wins in his/her first roll, congratulate the player and return with WON
   - If the user looses in his/her first roll, return with LOOSE and apologize
   - Let the user keep playing until (s)he wins / loses, give an appropriate massage and finish the game with the last roll value.
4) Your main( ) should
   - Call your function playGame( )
   - Ask the user if (s)he wants to continue playing another game, **keeping track of the numbers of Games lost and won**
   - When the user decides to finish playing, display the number of wins and losses (s)he had.
   - Give the user an appropriate message depending on the number of wins or losses (s)he had while playing
   - Return with a value of EXIT_SUCCESS
5) Your program **should use at least once the conditional operator**
6) Your program should present information to the user in clear way. **In the output given, a table is presented with the number of roll, the value of the first and second dice, the sum total of face values of the dice and the points that the user needs to achieve.**
7) Your program should compile with the flags -Wall -ansi -pedantic

Part B: Addition of a few Program Constraints (Bonus of 5Marks on successful completion):

Modify your program so the user can place a bet. Call your new program 02_Dice_B.c

- A user starts the game with $10
- A user can place a bet using multiple of $5
- If the user wins the game, it makes 3 times the bet
- If the user loses the game, it loses all the money he/she has bet
- You can decide if the user is able to place a bet every time it rolls the dice, or every time it plays a new game.
- The money won can get carried over to the next game
- If the bet money is 0, the user can not place any more bets.
- When the user decides to finish playing, the amount won should be displayed.

Deliverables:

1) Create a folder called algonquinUserID1_algonquinUserID2_A2 (e.g, "mynam00123_yourna45678_A2") that represents the two members of your team. Do all of your work in this folder and when complete, submit (**only one submission per group (if any)**), the zipped folder as per the "Lab Instructions" posted on Brightspace.
2) Write a program that will implement ALL the requirements, explicit and implicit, listed in the "Statement of the problem" above.
3) You must distribute your functions in a meaningful manner across multiple .c files. The .c files should contain functions that represent sensible groupings of functionality
4) You must define .h files as appropriate
5) Each function must have a header comments that explain what it does, and describe/explain its inputs (if any) and return value (if any)
6) You must **submit makefiles for both codes along with terminal screenshots and text files of your properly commented code**.

Marking:

This assignment is out of 40:

- 10 for coding correctness (i.e., correct results)
- 10 for appropriate and efficient logic (i.e., no spaghetti code!)
- 05 for sensible distribution of well-contained functions between files and makefile
- 15 for validation, proper error message and clear comments and coding convention